

Rapport Projet Info33

1. Choix techniques effectué

J'ai décider de créer une ferme et non une station de ski, car celle ci possède plus de structures que la station de ski, ce qui me permet de créer plusieurs structures différentes : une grange, une étable, un poulailler, ainsi qu'une maison

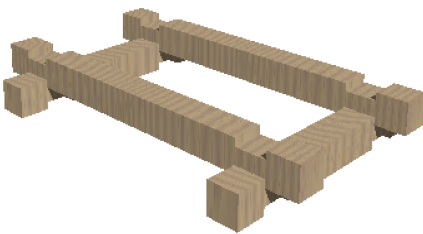
Je n'ai utilisé pratiquement que des macros pour effectuer toute les structures de la ferme, mais je pense qu'il aurait peut être été plus utile dans certains cas d'utiliser des #declare comme pour les sapins.

Piece de base : (coordonnée, translation, rotation, longueur, scale, texture)



Créé avec l'union de 4 box, 0.5 unité de largeur et de hauteur de base, mais peut s'agrandir avec le paramètre scale. Mon premier choix de largeur et de hauteur était de 1, mais j'ai préférer le diminuer car cela rendait un effet de mur trop épais. J'ai utilisé T_Wood7 du package « woods.inc » pour la texture de la piece de base.

Contour : (coordonnée, longueur, largeur, scale, texture)



Créer avec 4 piece de base, toute créer au point coordonnée, puis déplacer avec le paramètre translation :

```

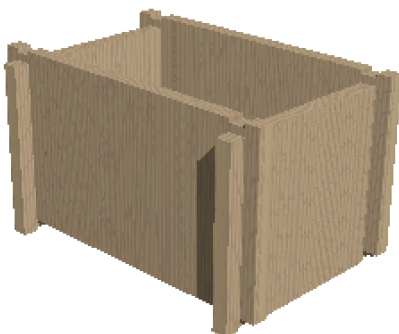
pieceDeBase( coor , < 0 , 0 , 0 > , 0 , larg ,sca,te)
pieceDeBase( coor , < 1 , -0.5 , 0.375 > , 90 * z , long ,sca,te)
pieceDeBase( coor , < 0 , long - 1.5 , 0 > , 0 , larg ,sca,te)
pieceDeBase(coor,< larg-0.5 ,-0.5,0.375> ,90*z , long , sca,te)

```

Je vais ensuite prendre l'exemple de la grange pour les murs et le pignon.

```
#macro grange(coor,larg,long,nbetage)
```

Pour créer les murs, j'ai tout simplement créer une boucle for avec comme paramètre le nombre d'étages,

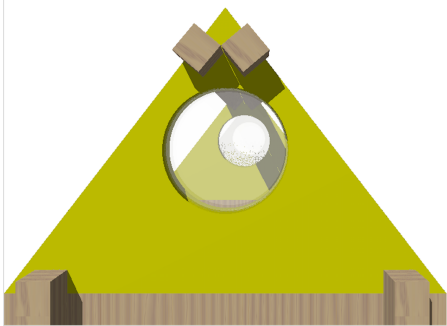


```

#for (i,0,nbetage,1)
    contour(<0,0,i/2>,long,larg,1,T_Wood7)
#end

```

Pignon : (coordonnée , longueur , hauteur , rotation , (boolean) fenetre)



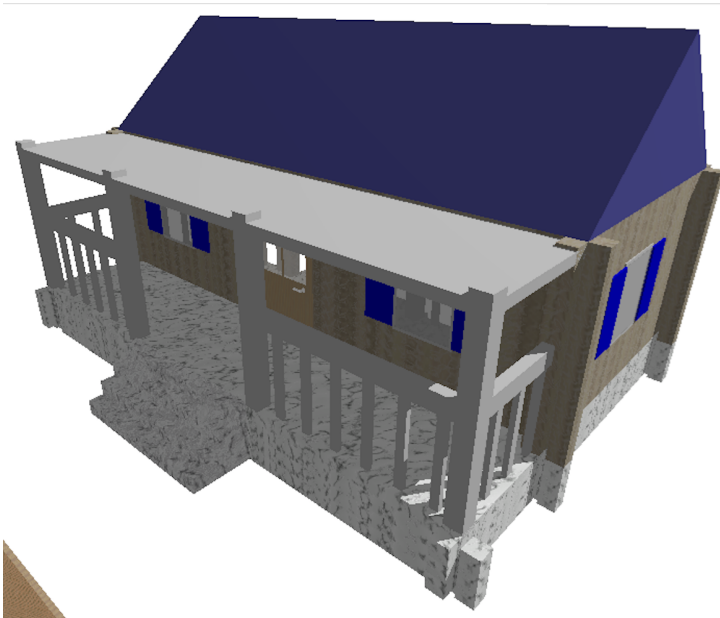
Pour la création d'une surface de révolution à partir d'une courbe, j'ai décidé de faire un sapin :

Afin de créer un environnement plus naturel, j'ai créé des sapins un peu partout sur la map, à l'aide de la fonction rand(sand(int)) :
Random = sand(1234);



```
#for (s,0,2000,i)
#local x1 = (100*rand( Random)* (-1+2* rand( Random)));
#local y1 = (100*rand( Random)* (-1+2* rand( Random)));
#if ((-27<x1)&(27>x1)&(-27<y1)&(27>y1))
#else
    object {
        sapin
        translate <x1,y1,0>
    }
#end
#end
```

La construction qui m'a pris le plus de temps pour les détails est la maison, je voulais vraiment créer une construction à ma manière quitte à dériver un peu du projet (comme ne pas faire de pignon).



2. Problèmes rencontrées

J'ai eu beaucoup de problème à effectuer ce projet, car j'avais beaucoup de zones sombres sur la programmation POV-ray, sur des choses que je ne comprenais pas, j'ai donc décidé de tester pas mal de choses avant de commencer comme la rotation suivant les axes x, y et z.

J'ai commencer par créer des macros pour tout les objets, mais j'ai ensuite vu que ce n'est pas très efficace pour imbriquer macros dans des objects, je ne pouvais donc pas par exemple compiler le code suivant :

```
#macro ligne(long, coor, tr, ro)
  object {
    #for(nb,1,long,1)
      maillage(coor, < 0 + nb * 0.82 - 1 , 0, 0 >+tr , ro)
    #end
  }
#end
```

Avec ce message d'erreur :

```
==== [Parsing...] =====
File 'test.pov' line 79: Parse Error: No matching } in 'object', union found
instead
Fatal error in parser: Cannot parse input.
Render failed
```

J'ai ensuite compris que la macro ne renvoyait pas, et n'était pas, un object mais créait tout simplement un object et l'affichait. Il fallait donc mettre en paramètres toutes les instructions secondaires que je voulais effectuer sur l'objet telles que la rotation, la texture...

Création du poulailler:

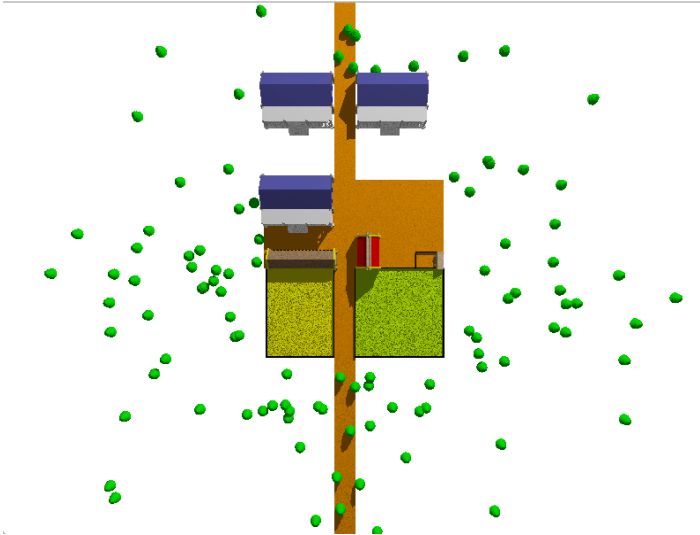
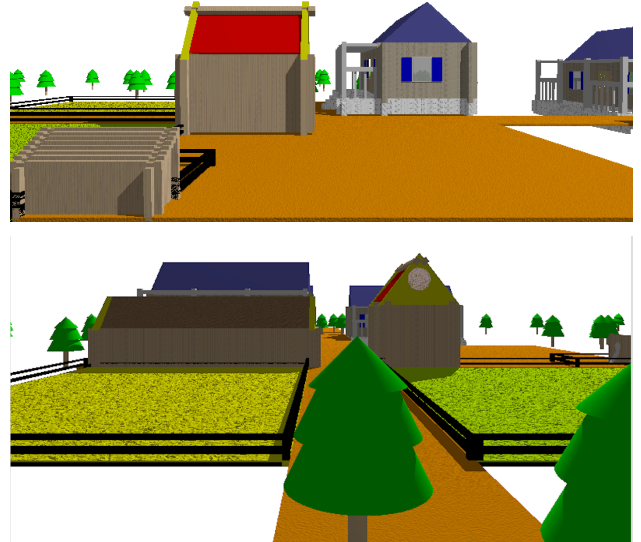
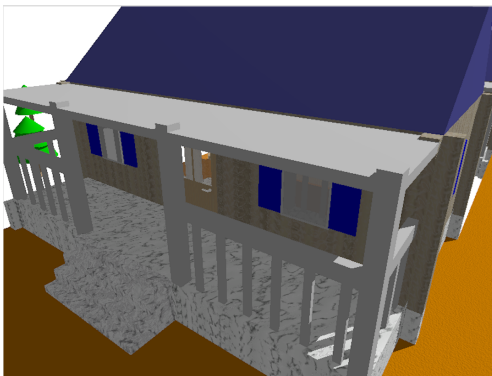
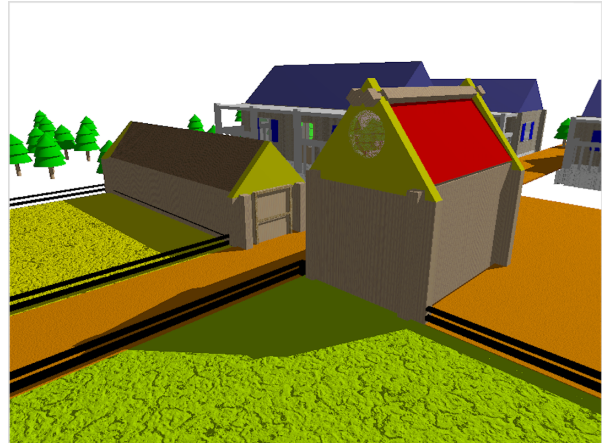
J'ai rajouté le paramètre scale dans la macro pieceDeBase à la création du poulailler. En effet, le poulailler étant plus petit que la grange ou la maison, il fallait le créer avec des murs plus petit, donc avec un paramètre scale de 0.5.

Création de la maison :

Les principales difficultés dans la maison ont été les fenêtres, comme je n'avais pas utilisé de #declare pour créer mes murs, je ne pouvais pas effectuer une différence {} pour créer des fenêtres transparente, j'ai donc créé des fenêtres miroir, ce qui au final donnait un bon rendu.

J'ai donc ensuite réutiliser cette technique pour les fenêtres de la porte. La poignée de la porte m'a demandé un peu plus d'imagination que le reste.

3. Captures d'écran du projet sous différents angles (Fond blanc)

CAMERA = 5**CAMERA = 1****CAMERA = 2****CAMERA = 10****CAMERA = 7**

4. Conclusion

Je n'étais pas très motivé à faire ce projet au début, car la synthèse d'image ainsi que la programmation POV-ray ne m'attirait pas vraiment en informatique, je ne voulais faire que de la programmation algorithmique du Java ou du C . Mais avec un peu de volonté, et en regardant les différentes créations en POV-ray ([Hall Of Fame](#)), j'ai finalement aimé découvrir ce style de programmation et créer des objets.